

Open Source
MANO

OSM Hackfest – Installation and first use

Gerardo García (Telefónica)

[Click HERE](#)

Before installing OSM

- Instructions

- https://osm.etsi.org/wikipub/index.php/LXD_configuration_for_OSM_Release_T_HREE

- Steps:

- Install lxd. Exec following commands as a non-root user:
 - `sudo apt-get update`
 - `sudo apt-get install -y lxd`
 - `newgrp lxd`
- Init LXD. Exec following command as a non-root user:
 - `sudo lxd init`
- Configure MTU of the default profile:
 - `sudo lxc profile device set default eth0 mtu <MTU>`
- Test LXD as a non-root user:
 - `lxc launch ubuntu:16.04 test`
 - `lxc exec test bash`
 - `root@test:~# apt-get update`

Installing OSM

- OSM installation from binaries:
 - `wget https://osm-download.etsi.org/ftp/osm-3.0-three/install_osm.sh`
 - `chmod +x install_osm.sh`
 - `./install_osm.sh`
- Other available options to install from:
 - `--lxdimages`
 - `--source`

After installing OSM

- Update RO to use the latest testing version
- Test OSM client
 - Add env variables to your .bashrc to use OSM client locally
 - Try 'osm'
- Test UI:
 - Google Chrome is the recommended browser
 - Access UI: `https:<IP_OSM>:8443`

Update RO to use the latest testing version

- ***NOTE: this is ONLY required for this Hackfest to have the latest working code in the RO***
- Instructions:
 - `dpkg -l |grep osm`
 - Modify `/etc/apt/sources.list`
 - `#deb http://osm-download.etsi.org/repository/osm/debian/ReleaseTHREE stable SO UI RO IM osmclient openvim`
 - `deb http://osm-download.etsi.org/repository/osm/debian/ReleaseTHREE-daily testing SO UI RO IM osmclient openvim`
 - `apt-get update`
 - `apt-cache show python-osm-ro`
 - `service osm-ro stop`
 - `apt-get install -y python-osm-ro`
 - `service osm-ro start`
 - `service osm-ro status`

For people using your own OSM

Test OSM client



- Add some env variables to your `.bashrc`:
 - `export OSM_HOSTNAME=<SO-ub_container_IP>`
 - `export OSM_RO_HOSTNAME=<RO_container_IP>`

- Try 'osm':

Usage: `osm [OPTIONS] COMMAND [ARGS]...`

Options:

`--hostname TEXT` hostname of server. Also can set `OSM_HOSTNAME` in environment

`--so-port INTEGER` hostname of server. Also can set `OSM_SO_PORT` in environment

...

For people using your own OSM Test UI



- Google Chrome installation (linux)
 - `wget -c https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb`
 - `sudo apt-get update && sudo apt-get install libappindicator1`
 - `sudo dpkg -i google-chrome-stable_current_amd64.deb`
- Configure OSM to allow access to the GUI from local web browser:
 - ***NOTE: this is ONLY required for this Hackfest (will be part of new version of installer)***
 - `sudo sysctl -w net.ipv4.conf.all.route_localnet=1`
 - `sudo iptables -t nat -A OUTPUT -p tcp -o lo --dport 8009 -j DNAT --to [IP LXC SO]:8009`
 - `sudo iptables -t nat -A POSTROUTING -s [IP HOST]/32 -d [IP LXC SO] -p tcp --dport 8009 -j MASQUERADE`
 - `sudo iptables -t nat -A OUTPUT -p tcp -o lo --dport 8443 -j DNAT --to [IP LXC SO]:8443`
 - `sudo iptables -t nat -A POSTROUTING -s [IP HOST]/32 -d [IP LXC SO] -p tcp --dport 8443 -j MASQUERADE`
 - Finally, add those NAT rules to `/etc/iptables/rules.v4` to make them persistent
- Test UI:
 - Open web browser: `https:<IP_OSM>://8443`

Local OSM instances @ETSI

- osm-demo: 172.21.1.3
- osm-plugtest1: 172.21.1.4
- osm-plugtest2: 172.21.1.5
- osm-hackfest1: 172.21.1.9
- osm-hackfest2: 172.21.1.10

For people using ETSI OSM servers Install OSM client



- Instructions in the main page for Rel THREE:
 - https://osm.etsi.org/wikipub/index.php/OSM_Release_THREE#Install_OSM_client
- Steps:
 - `curl http://osm-download.etsi.org/repository/osm/debian/ReleaseTHREE/OSM%20ETSI%20Release%20Key.gpg | sudo apt-key add -`
 - `sudo add-apt-repository -y "deb [arch=amd64] http://osm-download.etsi.org/repository/osm/debian/ReleaseTHREE stable osmclient"`
 - `sudo apt-get update`
 - `sudo apt-get install -y python-osmclient`

For people using ETSI OSM servers

Test OSM client



- Add some env variables to your `.bashrc`:

- `export OSM_HOSTNAME=<OSM_IP>`
- `export OSM_RO_HOSTNAME=<OSM_IP>`

- Try 'osm':

Usage: `osm [OPTIONS] COMMAND [ARGS]...`

Options:

`--hostname TEXT` hostname of server. Also can set `OSM_HOSTNAME` in environment

`--so-port INTEGER` hostname of server. Also can set `OSM_SO_PORT` in environment

...

For people using ETSI OSM servers

Test UI



- Google Chrome installation (linux)
 - `wget -c https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb`
 - `sudo apt-get update && sudo apt-get install libappindicator1`
 - `sudo dpkg -i google-chrome-stable_current_amd64.deb`

- Access UI:
 - `https://<IP_OSM>:8443`

Adding VIM accounts

- VIMs:

Name	Type	AUTH URL	tenant	user	Password	SDN controller
openstack1	openstack	http://172.21.2.20:5000/v2.0	xxx	xxx	xxx	N/A
openstack2-epa	openstack	http://172.21.2.22:5000/v2.0	xxx	xxx	xxx	N/A

- Test VIMs:

- ping <IP>
- curl http://<IP>:5000/v2.0
- Install python-openstackclient
 - sudo apt-get install python-openstackclient

Adding VIM accounts

- Load Openstack credentials and run some commands for testing:
 - export OS_AUTH_URL=xxx
 - export OS_USERNAME=xxx
 - export OS_TENANT_NAME=xxx
 - export OS_PASSWORD=xxx
 - openstack image list
 - openstack network list
 - openstack flavor list
 - openstack server list

Adding VIM accounts

- Add your first VIM 'openstack1' with the OSM client:
 - `osm vim-create --name openstack1 --account_type openstack \`
`--auth_url http://172.21.2.20:5000/v2.0 \`
`--user xxx --password xxx --tenant xxx \`
`--description "ETSI openstack site 1, with tenant xxx"`
 - `osm vim-list`

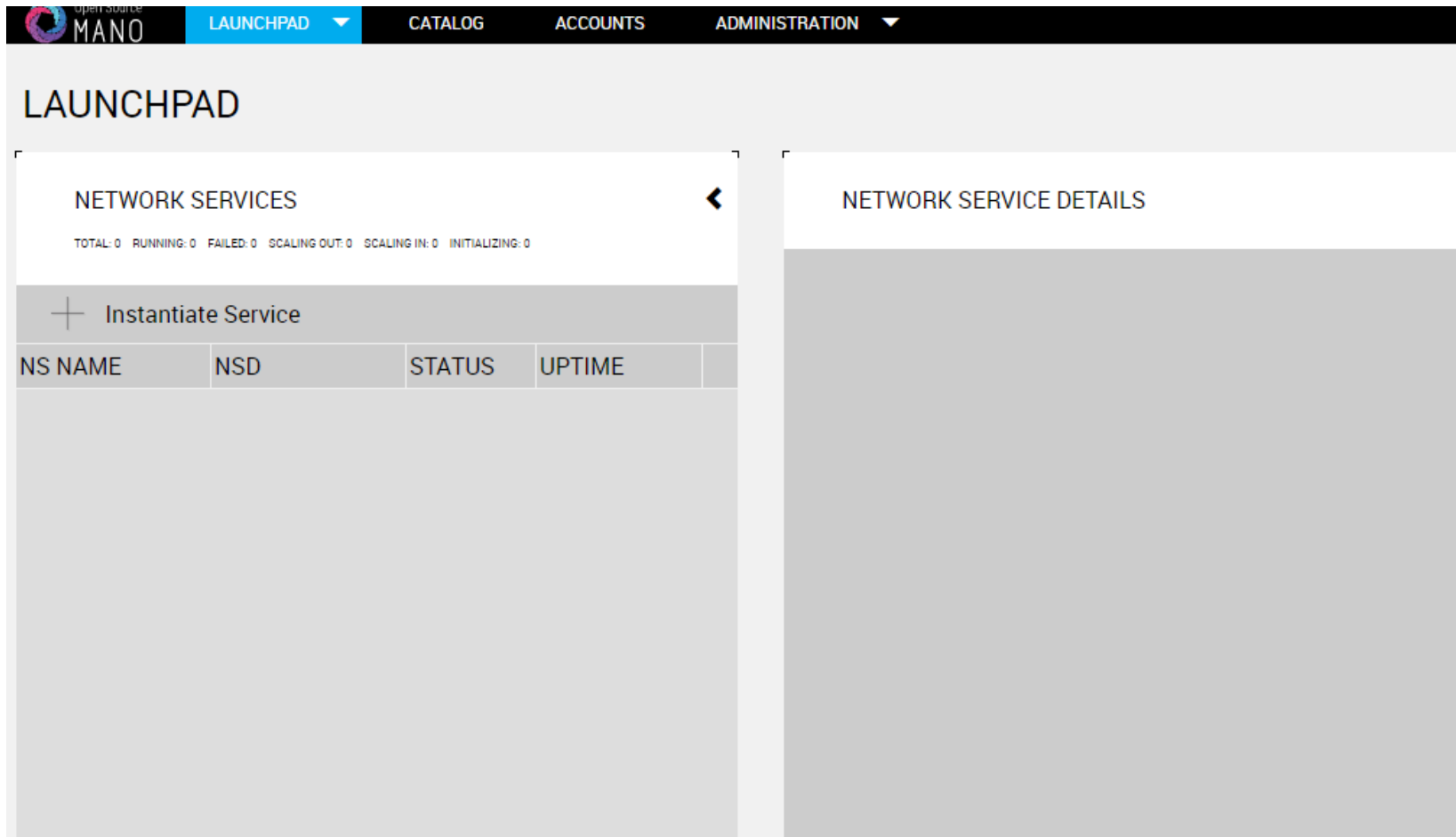
Adding images to the VIM

- Image management is not implemented in OSM today. It has to be done independently on each VIM.
- **IMAGES HAVE BEEN ALREADY ADDED TO THE REMOTE VIMS IN THE HACKFEST**
- Example for Openstack:
 - `openstack image create --file="./cirros-0.3.4-x86_64-disk.img" --container-format=bare --disk-format=qcow2 --public cirros034`

Adding images to the VIM

Image name in descriptors	Filename
ubuntu1604	xenial-server-cloudimg-amd64-disk1.img (you can get it from https://cloud-images.ubuntu.com/xenial/current/)
US1604	US1604.qcow2
hackfest3-mgmt	hackfest3-mgmt-qcow2
hackfest-pktgen	hackfest-pktgen-qcow2
cirros034	cirros-0.3.4-x86_64-disk.img

Deploying our first NS with OSM UI



The screenshot shows the Open Source MANO Launchpad interface. At the top, there is a navigation bar with the following items: 'open source MANO' logo, 'LAUNCHPAD' (highlighted in blue), 'CATALOG', 'ACCOUNTS', and 'ADMINISTRATION'. Below the navigation bar, the main content area is titled 'LAUNCHPAD'. On the left side, there is a panel titled 'NETWORK SERVICES' with a back arrow. Below this title, there is a status summary: 'TOTAL: 0 RUNNING: 0 FAILED: 0 SCALING OUT: 0 SCALING IN: 0 INITIALIZING: 0'. Below the status summary, there is a button labeled '+ Instantiate Service'. Below the button, there is a table with the following columns: 'NS NAME', 'NSD', 'STATUS', and 'UPTIME'. The table is currently empty. On the right side, there is a panel titled 'NETWORK SERVICE DETAILS' which is currently blank.

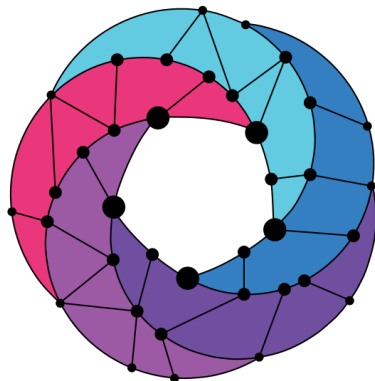
Deploying our first NS with the UI

- Add VNF package
- Add NS package
- Instantiate
- Get VNF record and obtain mgmt IP address
- Access to the VNF via SSH
- Delete NS instance
- Delete NS
- Delete VNF

Deploying our first NS with OSM client



- Add VNF and NS package
 - `osm upload-package cirros_vnf.tar.gz`
 - `osm vnfd-list`
 - `osm upload-package cirros_2vnf_ns.tar.gz`
 - `osm nsd-list`
- Instantiate
 - `osm ns-create --nsd_name cirros_2vnf_ns --ns_name <ns-instance-name> --vim_account <data-center-name>`
 - `osm ns-list`
- Delete NS instance
 - `osm ns-delete <ns-instance-name>`
 - `osm ns-list`
- Delete VNF and NS package
 - `osm nsd-delete cirros_2vnf_ns`
 - `osm nsd-list`
 - `osm vnfd-delete cirros_vnfd`
 - `osm vnfd-list`



Open Source
MANO

Find us at:

osm.etsi.org
osm.etsi.org/wikipub